IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: :
                                                   :
   Christopher Evans   :
                                                   :
Application. No.: TBD   : Group Art Unit: TBD
                                                   :
Filed: September 30, 2003   : Examiner: TBD

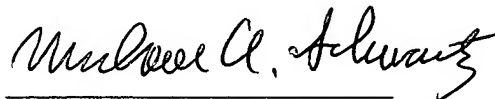Title: SQL JOIN ELIMINATION

CLAIM FOR PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

    A certified copy of corresponding Great Britain Application No. 0309975.1, filed April 30, 2003 is attached. It is requested that the right of priority provided by 35 U.S.C. 119 be extended by the U.S. Patent and Trademark Office.

                                             Respectfully submitted,

Date: September 30, 2003

Michael A. Schwartz (signature)
Michael A. Schwartz, Reg. No. 40,161
Swidler Berlin Shereff Friedman, LLP
3000 K Street, NW, Suite 300
Washington, DC 20007-5116
Telephone: (202) 424-7500
Facsimile: (202) 295-8478

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed   *A Evans.*

Dated   10 July 2003
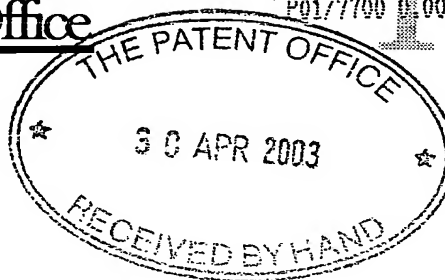
Patents Form 1/77

Patents Act 1977
(Rule 16)

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

01MAY03 E04037-16 00289

P01/7700 0 00-0309975.1

THE PATENT OFFICE

3 0 APR 2003

RECEIVED BY HAND

The Patent Office

Cardiff Road
Newport
South Wales
NP9 1RH

1.	Your reference	RSJ07764GB

2.	Patent application number *(The Patent Office will fill in this part)*	**0309975.1**

3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	Oracle International Corporation 500 Oracle Parkway M/S 5op7, Redwood Shores California 94065 USA
	Patents ADP number *(if you know it)*	837301001
	If the applicant is a corporate body, give the country/state of its incorporation	DELAWARE, USA

4.	Title of the invention	SQL Join Elimination

5.	Name of your agent *(if you have one)*	Gill Jennings & Every
	"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*	Broadgate House 7 Eldon Street London EC2M 7LH
	Patents ADP number *(if you know it)*	745002

		Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*
6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number			

		Number of earlier application		Date of filing *(day / month / year)*
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application			

8.	Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:* a) *any applicant named in part 3 is not an inventor, or* b) *there is an inventor who is not named as an applicant, or* c) *any named applicant is a corporate body.* *See note (d))*	YES

Patents Form 1/77

Patents Form 1/77

9. Enter the number of sheets for any of the
    following items you are filing with this form.
    Do not count copies of the same document

<table>
<tr><td></td><td colspan="2">Continuation sheets of this form</td></tr>
<tr><td></td><td>Description</td><td>9</td></tr>
<tr><td></td><td>Claim(s)</td><td>2</td></tr>
<tr><td></td><td>Abstract</td><td>-</td></tr>
<tr><td></td><td>Drawing(s)</td><td>4</td></tr>
</table>

10. If you are also filing any of the following,
    state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	1
Any other documents *(please specify)*	NO

11.	For the applicant Gill Jennings & Every	I/We request the grant of a patent on the basis of this application. Signature ~~Michael Lord~~ Date 30 April 2003

12.	Name and daytime telephone number of person to contact in the United Kingdom	R.E. Skone James 020 7377 1377

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

SQL JOIN ELIMINATION

This invention relates to a method of preventing execution of unnecessary joins between tables in a database
5   referred to by a Structured Query Language (SQL) statement.

The concept of a join between tables in a database is well known. A simple example will be described here with reference to Figure 1. This shows two tables with the names EMP and DEPT. The first table, EMP, lists the names
10   of the employees of a company under the column ENAME against the number of the department for which they work under the column DEPTNO. The table DEPT has a similar column named DEPTNO in which each department number is only listed once and adjacent to this is a column entitled DESC
15   giving the names of the respective departments. It can be seen that a many to one relationship exists between the tables EMP and DEPT via their respective DEPTNO columns. That is to say that a value in the DEPTNO column of DEPT can only appear once whilst the same value can appear many
20   times in the DEPTNO column of EMP. In this context, table EMP is referred as the detail table and table DEPT is referred to as the master table.

The necessity for a join between these two tables comes about if, for example, it was desired to extract the
25   names of the employees and their respective department names.   A suitable structured query language (SQL) statement to perform this function is:

     SELECT ENAME, DNAME FROM EMP, DEPT
30   WHERE EMP.DEPTNO = DEPT.DEPTNO

This statement informs the database that values in the DEPTNO column of table EMP can be considered equivalent to values in the DEPTNO column of table DEPT and allows it to
35   return the results to the user, correctly indicating that Chris and Steve work in the R&D Department and that Paul works in the Sales Department.

Typically, a view is used to present the two joined
tables to a user as a single table. For example, a view
called EMPDEPT may be defined as the SQL statement given
above. Subsequent SQL statements can then be executed that
5   refer to the view EMPDEPT. For example, the SQL statement:

SELECT ENAME, DNAME FROM EMPDEPT

will return the same results as the previous SQL statement.
10 However, a problem exists in that, under certain
circumstances, the join will be executed even though it is
not needed. For example, the SQL statement:

SELECT ENAME FROM EMPDEPT

15
will still result in the join being executed although, in
this instance, it is not necessary to execute the join to
retrieve the desired data.

The execution of joins in situations such as this
20 where they are unnecessary is extremely undesirable since
they can be very costly in terms of processing speed,
especially as the size of the database increases. Clearly,
there exists a need for a method of preventing such
unnecessary joins from being executed.

25   In accordance with a first aspect of the present
invention, there is provided a method of preventing
execution of unnecessary joins between tables in a
database, the method comprising the steps of:

    a.   presenting a Structured Query Language (SQL)
30   statement to the database, the SQL statement
           having a scope that extends to a set of tables
           in the database and returning a set of results
           from the database;

    b.   preparing a list of tables that are within the
35   scope of the SQL statement but that are not
           referred to by the SQL statement;

c.   removing tables that must be accessed in order to return the set of results from the list in accordance with a predetermined set of rules; and,

5   d.   preventing execution of joins involving any of the tables remaining in the list.

Hence, this invention prevents the execution of unnecessary joins in a situation where an SQL statement includes one or more joins but the required data can be

10   extracted from a subset of the joined tables. Prior to this invention, the join or joins would be executed irrespective of whether data was required from all tables or not.

Typically, the predetermined set of rules includes a

15   rule allowing removal of a table from the list if this table is part of a join chain on a master table.

Normally, the predetermined set of rules includes a rule allowing removal of a table from the list if this table forms the detail table in a join between a master

20   table and a detail table.

Typically, the predetermined set of rules includes a rule allowing removal of a table from the list if detail item values might not exist in a master table joined to a detail table.

25   Preferably, the predetermined set of rules includes a rule allowing removal of a table if that table has a mandatory filter.

In the event that the removal of a table from the list would normally be allowed since detail item values might

30   not exist in a master table joined to a detail table or since that table has a mandatory filter then, preferably, the predetermined set of rules further includes a rule that prevents removal of a table from the list if the join is an outer join on a master table.

35   The invention will typically be provided as a computer program comprising computer program code means adapted to

perform the steps of the first aspect of the invention when said program is run on a computer.

Further, there may be provided a computer program product comprising program code means stored on a computer readable medium for performing a method according to the first aspect of the invention when said program product is run on a computer.

An embodiment of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1 shows two tables in a database;

Figure 2 shows a flowchart for a method according to the invention;

Figure 3 shows three tables in a join chain; and,

Figure 4 shows a table in which not all of the detail item values exist in the master table.

As already mentioned, Figure 1 shows two tables, EMP and DEPT, in a database. The tables are related by their respective DEPTNO columns. In order to simplify the presentation of information to a user, a view EMPDEPT may be defined as shown below:

```
SELECT ENAME, DNAME FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
```

In order to prevent the execution of unnecessary joins, the method shown in the flowchart in Figure 2 is used. The first step 1 in this process generates a list of tables that are not referred to by an SQL statement but that are within its scope. For example, considering the view EMPDEPT already defined, the SQL statement:

```
SELECT ENAME FROM EMPDEPT
```

Both the tables EMP and DEPT are within the scope of the SQL statement although the statement only refers directly to the table EMP. It does not refer directly to the table DEPT. Hence, using the simple example shown in

Figure 1 having only these two tables and taking this SQL statement the list of tables will consist merely of DEPT. This can be considered to be a list of candidates for which it may be possible to prevent the execution of a join. In order to ascertain whether it is possible to prevent the execution of a join, the decision making steps 3 to 7 shown in Figure 2 must be performed.

Step 2 of the process takes the first table from the list for processing by steps 3 to 7. In this example, this table is DEPT. This first table is then subjected to the decision making process of steps 3 to 7. These can be considered as a set of rules that must be satisfied in order for the table to remain in the list. Execution of a join involving any of the tables remaining in the list after the process has been completed will be prevented.

The first decision step 3 examines whether the table is part of a join chain. This concept is best described with respect to an example which is shown in Figure 3. In this example, the tables EMP and DEPT have been set out as before although DEPT now has a further column known as LOC in which the location of each department is listed. A further table known as GEOGRAPHY has a corresponding LOC column and a CURRENCY column indicating the currency in use at that location. These tables may form a join chain in which the tables EMP and GEOGRAPHY are joined via table DEPT. For example, a view known as EMPDEPTLOC may be defined as:

```
SELECT ENAME, DNAME, CURRENCY
FROM EMP, DEPT, GEOGRAPHY
WHERE EMP.DEPTNO = DEPT.DEPTNO
AND DEPT.LOC = GEOGRAPHY.LOC
```

Then, the SQL statement:

```
SELECT ENAME, CURRENCY FROM EMPDEPTLOC
```

will return the following results:

CHRIS   £

STEVE   £

5   PAUL   . $

This statement does not directly refer to table DEPT
and hence, table DEPT would be listed as a candidate for
which a join need not be executed. However, if the join is
10   not executed, this would lead to incorrect results since it
is necessary to maintain the join chain between EMP and
GEOGRAPHY through DEPT. Decision step 3 will determine
that table DEPT is in a join chain and it will then be
removed from the list in step 8, thereby ensuring that the
15   join in the previous SQL statement will be executed. If,
however, the table is not in a join chain the process
continues to step 4.

Step 4 in the process is used to remove a table from
the list if this table forms the detail table in a join
20   between a master table and a detail table. That is to say
that, if the join has a one to many or many to one
relationship and the table is on the "many" end of the join
then the table must be removed from the list so that the
join is executed. The table on the "many" end of the join
25   is known as the detail table whilst that on the "one" end
of the join is the master table.

An example of a many to one relationship can be seen
in Figure 1 in which the column DEPTNO in table EMP has
many instances of the value 10 for example whilst each
30   value only appears once in column DEPTNO of table DEPT.
This is a many to one join between these two tables in
which DEPT is the master table and EMP is the detail table.

The SQL statement:

SELECT DNAME FROM EMPDEPT

35

should give the results shown below:

R&D

R&D

SALES

5   However, an incorrect implementation that eliminated
the join between the two tables would give the following
results:

R&D

10   SALES

ACCOUNTS

In accordance with step 4, if the table is a detail
table in a join between the master table and the detail
table, then it must be removed from the list and this is
15   performed by step 8. If however this condition is not met
then processing proceeds to step 5.

Step 5 is used to bypass steps 6 and 7 in the event
that a join is an outer join.

An outer join is best described by way of example.
20   The tables EMP and DEPT of Figure 4 are similar to those of
Figure 1 except that a value of "JOHN" has been added under
the column ENAME with a DEPTNO value of "40". This value
of 40 has no corresponding entry in the DEPT table but an
outer join, as shown by the following SQL statement:

25

SELECT ENAME FROM EMPDEPT

where EMPDEPT is a view defined as:

30   SELECT ENAME, DNAME FROM EMPDEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO (+)

will still return the value of "JOHN".

The outer join operator "(+)" informs the database to
35   return the value of JOHN even though he has no department.
An outer join returns all rows from the table without the

outer join operator for which there are no matching rows in a table with the outer join operator.

If the join is an outer join then step 5 determines that it is not necessary to proceed with steps 6 and 7 and the table is not removed from the list by step 8, processing proceeding instead to step 9.

The next step 6 determines whether detail item values always exist in a master table joined to a detail table or otherwise. For example, considering Figure 4 again, EMP is a detail table joined to a master table DEPT. If detail item values must always exist in the master table, then the join between EMP and DEPT need not be executed for the SQL statement:

    SELECT ENAME FROM EMPDEPT

However, if detail item values might not exist in the master table, as is shown in Figure 4, then the join must be executed and the table is removed from the list by step 8.

If processing proceeds to step 7 then this step determines whether the table has a mandatory filter attached to it. For example, this filter might be used to return results for a query to the ENAME column only where the corresponding DNAME value is SALES.

In order to implement this mandatory filter, a view known as EMPDEPTSALES may be defined as:

    SELECT ENAME, DNAME FROM EMP, DEPT
    WHERE EMP.DEPTNO = DEPT.DEPTNO
    AND DNAME ! = `SALES'

The mandatory filter is invoked by the "DNAME ! = `SALES'" fragment of this definition. In the SQL statement:

    SELECT ENAME FROM EMPDEPTSALES

the join cannot be removed because of the mandatory filter on table DEPT.

If the table does have a mandatory filter then the join must be executed and the table is removed from the list by step 8. Otherwise, if there is no mandatory filter, then the join can be removed and the table remains in the list.

In any event, processing eventually proceeds to step 9 as shown in Figure 2 which determines if the current table is the last table in the list. If it is not then processing proceeds to step 10 which takes the next table from the list and returns to step 3 to consider this next table. If, however, this is the last table then the process ends.

Execution of a join will be prevented if that join involves any of the tables remaining in the list after proceeding through the flowchart shown in Figure 2.

It is important to note that while the present invention has been described in a context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of a particular type of signal bearing media actually used to carry out distribution. Examples of computer readable media include recordable-type media such as floppy disks, a hard disk drive, RAM and CD-ROMs as well as transmission-type media such as digital and analogue communications links.

CLAIMS

1. A method of preventing execution of unnecessary joins between tables in a database, the method comprising the steps of:

    a. presenting a Structured Query Language (SQL) statement to the database, the SQL statement having a scope that extends to a set of tables in the database and returning a set of results from the database;

    b. preparing a list of tables that are within the scope of the SQL statement but that are not referred to by the SQL statement;

    c. removing tables that must be accessed in order to return the set of results from the list in accordance with a predetermined set of rules; and,

    d. preventing execution of joins involving any of the tables remaining in the list.

2. A method according to claim 1, wherein the predetermined set of rules includes preventing removal of a table from the list if this table is part of a join chain.

3. A method according to either of the preceding claims, wherein the predetermined set of rules includes a rule allowing removal of a table from the list if this table forms the detail table in a join between a master table and a detail table.

4. A method according to any of the preceding claims, wherein the predetermined set of rules includes a rule allowing removal of a table from the list if detail item values might not exist in a master table joined to a detail table.

5. A method according to any of the preceding claims, wherein the predetermined set of rules includes a rule allowing removal of a table from the list if that table has a mandatory filter.

6. A method according to claim 4 or claim 5, wherein the predetermined set of rules further includes a rule preventing removal of a table from the list that would otherwise be allowed, if the join is an outer join on a master table.

7. A computer program comprising computer program code means adapted to perform the steps of any of the preceding claims when said program is run on a computer.

8. A computer program product comprising program code means stored on a computer readable medium for performing the method of any claims 1 to 6 when said program product is run on a computer.

9. A method substantially as hereinbefore described with reference to the accompanying drawings.

EMP

ENAME	DEPTNO
CHRIS	10
STEVE	10
PAUL	20

DEPT

DEPTNO	DNAME
10	R & D
20	SALES
30	ACCOUNTS

Figure 1

Figure 2

START

1 — Generate list of tables not referred to by SQL statement

2 — Take first table in list

3 — Is table in a join chain?
- N
- Y

4 — Is table a detail table?
- N
- Y

5 — Is join an outer join?
- Y
- N

6 — Do detail item values always exist in master table?
- Y
- N

7 — Does table have mandatory filter?
- Y
- N

8 — Remove table from list

9 — Is this last table in list?
- Y → END
- N

10 — Take next table in list

EMP
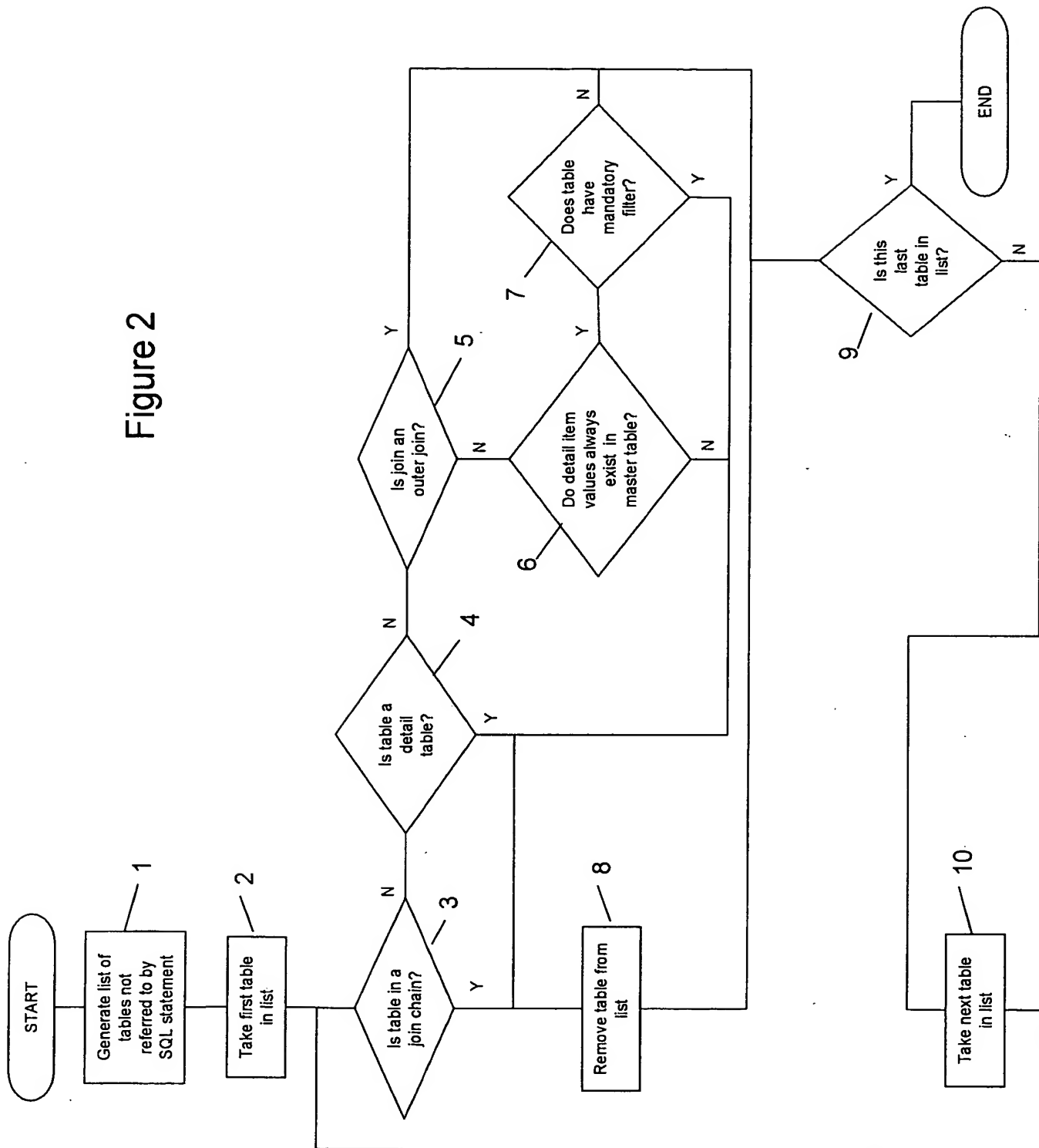
ENAME	DEPTNO
CHRIS	10
STEVE	10
PAUL	20

DEPT

DEPTNO	DNAME	LOC
10	R & D	UK
20	SALES	US
30	ACCOUNTS	US

GEOGRAPHY

LOC	CURRENCY
UK	£
US	$

Figure 3

EMP

ENAME	DEPTNO
CHRIS	10
STEVE	10
PAUL	20
JOHN	40

DEPT

DEPTNO	DNAME
10	R & D
20	SALES
30	ACCOUNTS

Figure 4